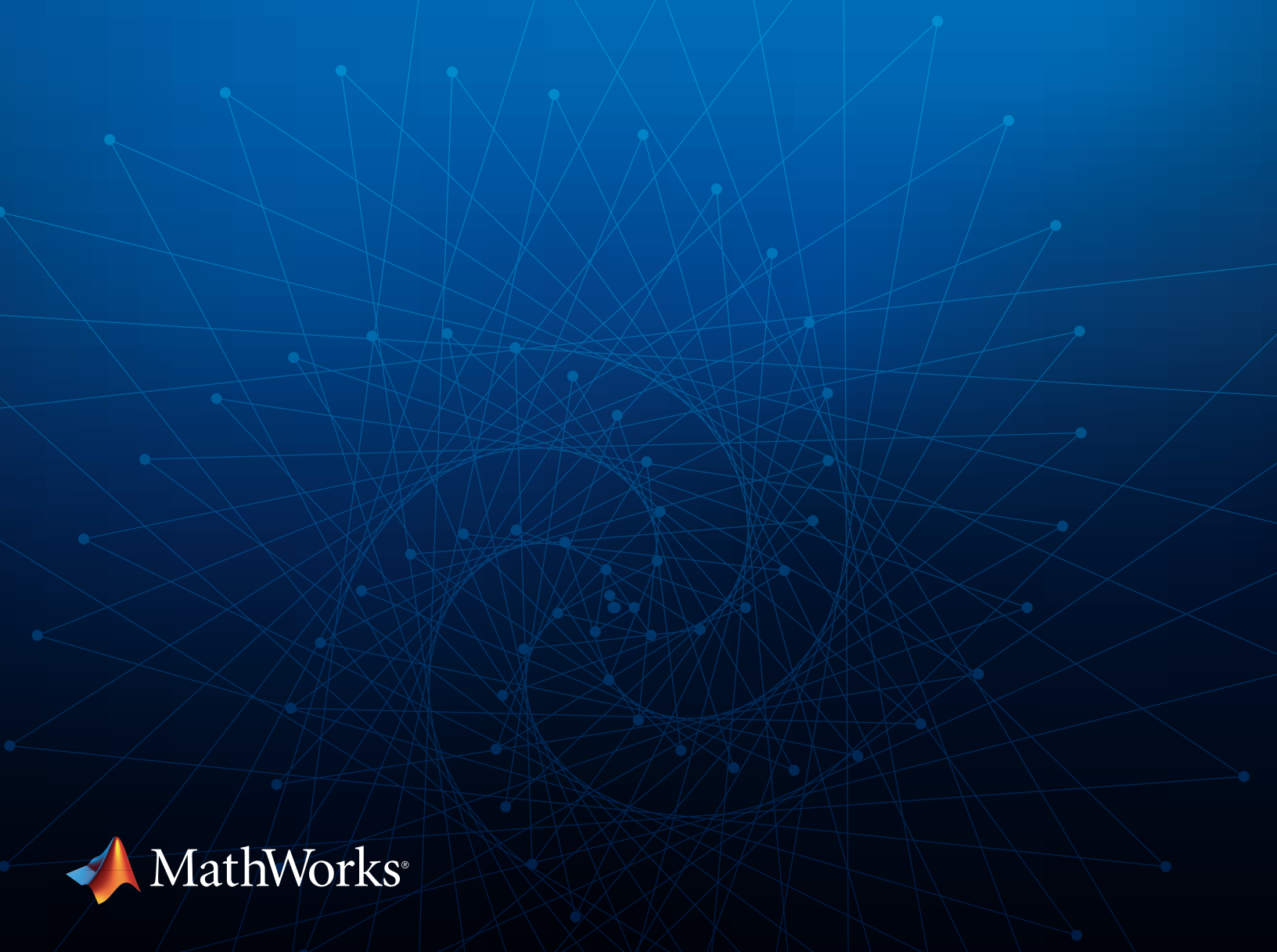
WHITE PAPER

Cómo medir la rentabilidad sobre la inversión del diseño basado en modelos

A medida que aumenta la complejidad de los sistemas ciberfísicos, resulta cada vez más difícil mantener la calidad y controlar los costes con los enfoques tradicionales de desarrollo de sistemas. Para hacer frente a este desafío y mejorar su posición competitiva, las empresas están adoptando el diseño basado en modelos para el desarrollo de sistemas y software. No obstante, las ventajas de adoptar el diseño basado en modelos y los procesos que requiere para que se pueda aprovechar plenamente deben estar justificados antes de realizar la inversión. El marco de rentabilidad sobre la inversión del diseño basado en modelos que se describe en este white paper proporciona una herramienta analítica para justificar la inversión en el diseño basado en modelos sirviéndose de la cuantificación del ahorro esperado por el uso del diseño basado en modelos en lugar de un enfoque de desarrollo tradicional.



MathWorks®

Introducción

A medida que aumentan el alcance y la complejidad de los requisitos de los clientes, ha ocurrido lo propio con la lógica y el software de control de los sistemas. Mientras desarrollan los millones de líneas de código necesarias para aviones y automóviles con plazos cada vez más ajustados, las organizaciones se dan cuenta de que los procesos de desarrollo tradicionales no son suficientes para cumplir con los objetivos de calidad y plazos. El diseño basado en modelos para el desarrollo de sistemas reduce los costes identificando los errores en una etapa inicial del proceso de desarrollo y disminuyendo el número total de errores latentes. El diseño basado en modelos proporciona una ventaja competitiva, dado que ofrece maquinaria industrial, robótica, sistemas inalámbricos y otros sistemas complejos con mayor calidad, a un menor coste y en menos tiempo.

Proceso de desarrollo tradicional frente a diseño basado en modelos

En un proceso de desarrollo tradicional, las tareas de cada etapa se ejecutan secuencialmente en diferentes entornos de herramientas, con muchos pasos manuales que provocan dificultades (Figura 1). En cada etapa de desarrollo, desde la definición de los requisitos hasta el funcionamiento del sistema, se crean ineficiencias cuando no se adopta un enfoque de diseño basado en modelos.

Los requisitos del sistema se capturan a partir de texto, con herramientas como Microsoft® Word® o IBM® Engineering Requirements Management Doors®, mientras que las arquitecturas del sistema se especifican en herramientas de representación gráfica, lo que dificulta su análisis, interpretación y gestión a medida que se realizan cambios. Los diseños de subsistemas se crean con herramientas específicas del dominio, lo que impide realizar pruebas en el nivel de sistema hasta después de la implementación en software o hardware. Luego, los diseños se convierten manualmente a código, un proceso lento y propenso a errores. Si bien se generan errores en cada fase, es la fase de pruebas donde se acumulan todos los errores que se han generado en las fases anteriores. La falta de un entorno de herramientas común, la gran cantidad de pasos manuales y de errores encontrados en la última etapa aumentan la duración y el coste de desarrollo. Los datos operativos y la información resultante tampoco se utilizan de manera eficaz, con lo que se pierde la posibilidad de incorporar valiosas mejoras en la eficiencia y el tiempo de actividad del sistema mientras está en funcionamiento.

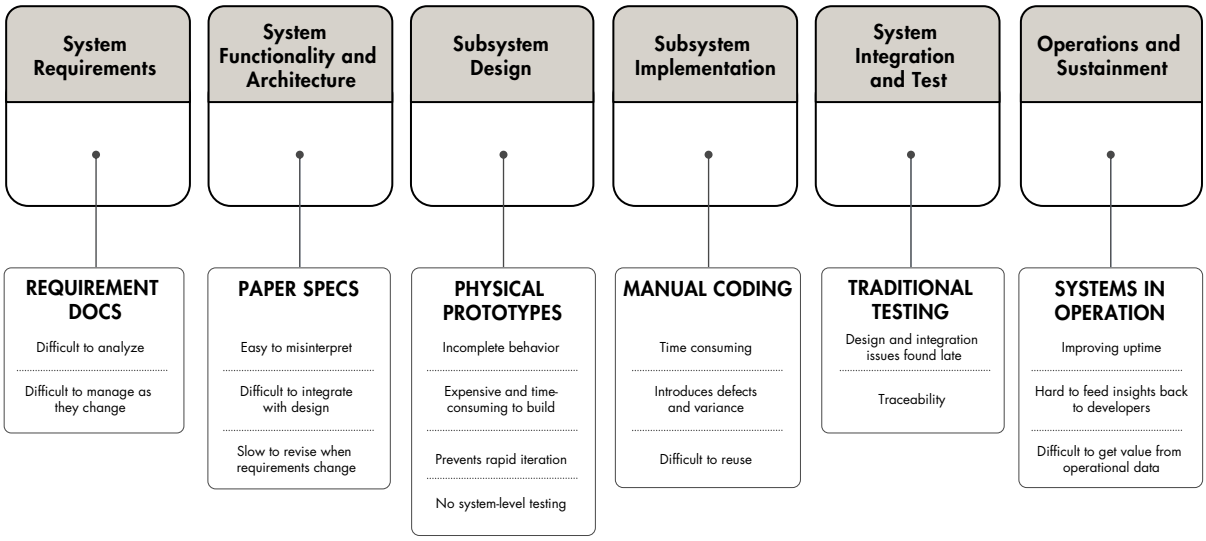


Figura 1. Proceso tradicional de desarrollo de software. Este enfoque requiere muchos pasos manuales innecesarios que pueden causar errores.

El diseño basado en modelos (Figura 2) hace un uso sistemático de modelos a lo largo de todo el proceso de desarrollo. Comienza con el mismo conjunto de requisitos del sistema que un proceso tradicional. Sin embargo, en lugar de servir como base para especificaciones textuales, los requisitos se utilizan para desarrollar una arquitectura del sistema que es una especificación ejecutable en forma de modelos de comportamiento y arquitectura. El equipo de ingeniería utiliza esos modelos de arquitectura para aclarar los requisitos y las especificaciones. Así, los modelos se utilizan como base para desarrollar un diseño detallado de subsistemas.

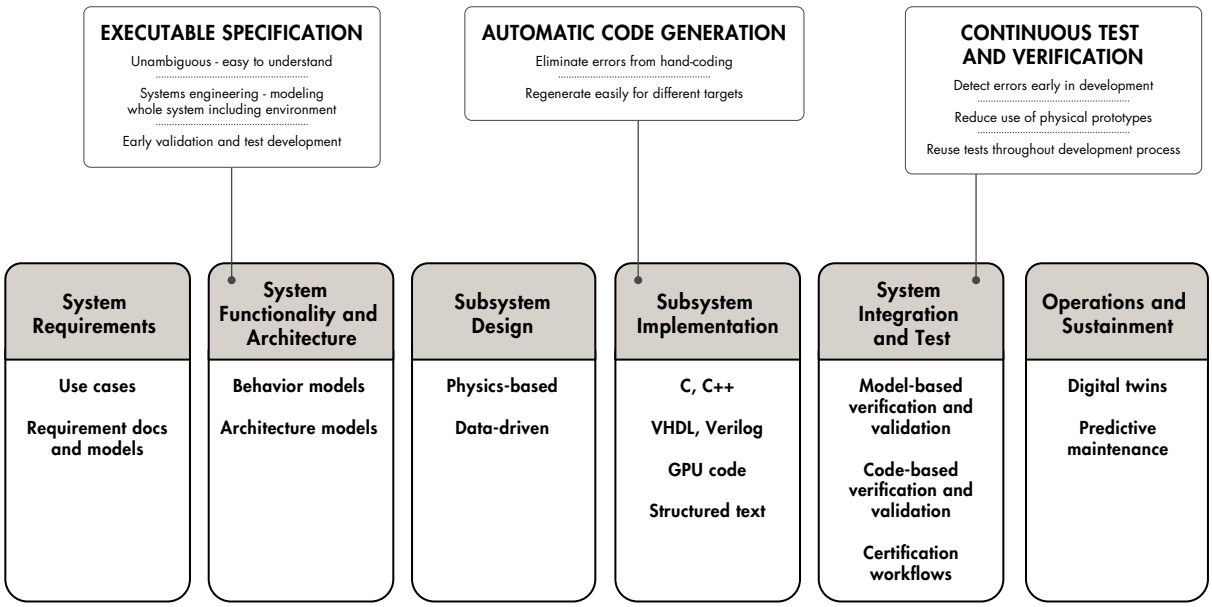


Figura 2. Proceso de desarrollo de software utilizando el diseño basado en modelos. Este enfoque emplea un modelo en el nivel de sistema como especificación ejecutable durante todo el desarrollo, y permite el diseño y la simulación en los niveles de sistema y componente, la generación de código automática, y la realización continua de pruebas y verificación.

Con MATLAB® y Simulink® para el diseño basado en modelos, se puede simular el diseño en el nivel de sistema y detectar errores en la interfaz antes de la implementación. Una vez que se ha concluido el diseño, se genera automáticamente código con calidad de producción y casos de prueba a partir de los modelos, lo que elimina los errores de código manual. Con este flujo de trabajo, se puede permanecer en el mismo entorno desde la definición de requisitos hasta las pruebas del sistema, lo que minimiza el volumen de trabajo manual. Además, las pruebas pueden comenzar en la fase de requisitos, cuando se simulan las especificaciones ejecutables de modelos para verificar que se cumplen los requisitos. Como resultado, los errores se detectan y eliminan antes, lo que reduce el coste total de desarrollo.

Ahorros en ingeniería de sistemas

El uso sistemático de modelos en la ingeniería de sistemas basada en modelos puede suponer hasta un 55% de ahorro total al cabo de dos años, según un estudio realizado por Jerry Krasner para Embedded Market Forecasters [1]. El uso de modelos reduce la captura de requisitos a partir de texto, que suelen especificarse en exceso debido a errores de comunicación anteriores. Con el diseño basado en modelos, los modelos sirven como lenguaje común durante todo el proceso de desarrollo. Este enfoque reduce la ambigüedad de las especificaciones del producto y permite el uso de simulaciones para validar los requisitos. El uso de modelos para simular y ajustar las especificaciones del producto antes de crearlo es el motivo principal de los ahorros.

El diseño basado en modelos también permite explorar varias soluciones al desarrollar requisitos y arquitecturas del sistema, lo que a su vez permite realizar iteraciones de diseño más rápidas y lograr consenso en una etapa inicial del proceso de diseño. En el método basado en modelos, se utiliza una sola cadena de herramientas desde la fase de ingeniería de sistemas y durante las fases de desarrollo y operación de un proyecto, lo que permite conectar las arquitecturas de sistemas a los diseños de sistemas de manera fluida y eficiente.

Ahorros en desarrollo

Las organizaciones que adoptan el diseño basado en modelos obtienen un ahorro de entre el 20 y el 60% respecto de los métodos tradicionales [2, 3]. El grueso de este ahorro proviene de un mejor análisis de los requisitos combinado con la realización de pruebas y verificación en etapas iniciales y continuas. Al simular los requisitos y diseños con modelos, los errores se detectan mucho antes en el proceso de desarrollo, cuando el coste de corregirlos es inferior en órdenes de magnitud (Figura 3).

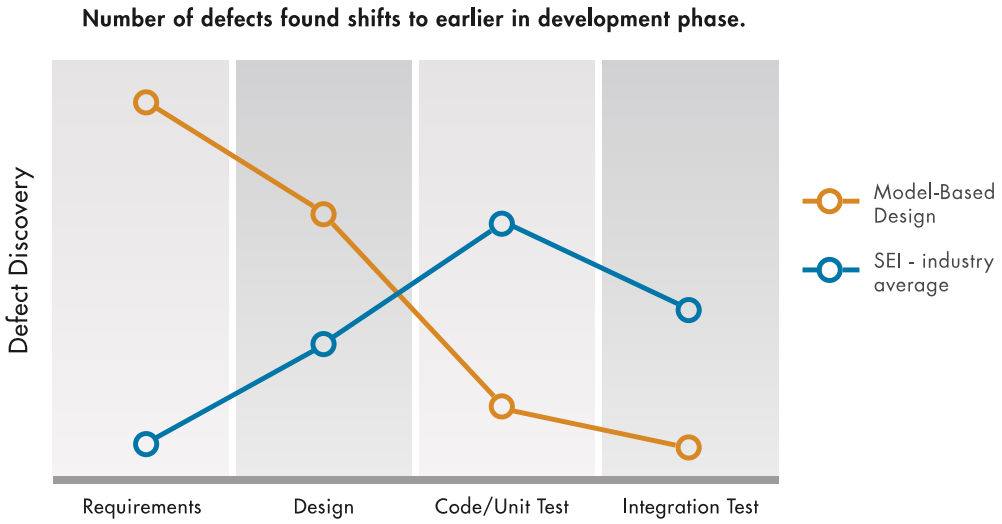


Figura 3. Errores identificados en la fase de desarrollo. El diseño basado en modelos adelanta la detección de errores durante el proceso.

Cuantificación de ahorros con el marco de ROI del diseño basado en modelos

El marco de la rentabilidad sobre la inversión está diseñado para estimar ROI que genera adoptar el diseño basado en modelos en proyectos específicos. Tomando como base el tamaño del proyecto, el tamaño del equipo de trabajo y otros factores, el marco calcula el coste de desarrollo tradicional utilizando el modelo constructivo de costes (COCOMO), y luego resta el ahorro que genera el diseño basado en modelos para obtener el coste de desarrollo correspondiente al diseño basado en modelos. Se seleccionó el modelo COCOMO básico para el marco porque es una herramienta de estimación de costes paramétricos general cuyo uso está muy extendido en el sector aeroespacial y de defensa, en el que la responsabilidad respecto de los costes de adquisición exige modelos rigurosos para la estimación de costes de software.

La rentabilidad sobre la inversión se calcula teniendo en cuenta el ahorro que puede suponer utilizar el diseño basado en modelos para el desarrollo. Este cálculo utiliza métricas del Instituto de Ingeniería de Software (SEI), el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) y estudios de diferentes industrias [1, 4, 5, 6]. Dado que el alcance de los proyectos, los procesos existentes y la experiencia de los equipos de trabajo con el diseño basado en modelos varía según las industrias y empresas, el marco de rentabilidad sobre la inversión se puede personalizar para proyectos y equipos específicos.

Tomemos como caso de referencia un proyecto de software con 50.000 líneas de código. Aplicando el modelo COCOMO básico, el coste de desarrollo empleando métodos tradicionales sería de aproximadamente 6 millones de dólares (USD). Para calcular los ahorros del diseño basado en modelos con respecto a los métodos tradicionales, cada fase de desarrollo (requisitos, diseño, implementación y pruebas) se analiza en función de métricas de la industria. Luego, se resumen los ahorros y se resta del coste tradicional de desarrollo. En este caso, el coste del diseño basado en modelos es de 3 millones de dólares (USD), un ahorro del 50% en comparación con el método tradicional.

Para llegar al 50% de ahorro, el marco examina las ineficiencias del proceso de desarrollo tradicional que el diseño basado en modelos elimina, y calcula el ahorro en función de datos empíricos procedentes de clientes, entrevistas con clientes, métricas de la industria y medias. El ahorro correspondiente a cada fase de desarrollo se calcula por separado, con objeto de que el marco pueda adaptarse a una adopción paso a paso del diseño basado en modelos.

Ahorros durante el desarrollo de requisitos

Para hacerse una idea de cómo funciona el marco, observemos una ineficiencia de la fase de requisitos. Utilizar modelos para detectar requisitos imprecisos, incoherentes o no comprobables permite detectar un mayor porcentaje de errores. El caso de referencia presupone que el 15% de los requisitos contienen errores o necesitan reelaboración; esta cifra es una estimación obtenida a partir de entrevistas de la industria. Detectar estos errores en la fase de requisitos significa evitar el coste que supone una reelaboración posterior durante la fase de desarrollo. Parte del ahorro en requisitos se obtiene multiplicando el número de requisitos con errores por el tiempo promedio en resolver aquellos que se detectan después de la fase de requisitos. En el caso de referencia, la media por cada error de requisito es de 4,5 horas [6]. Según este cálculo, el diseño basado en modelos ahorra 3.375 horas de ingeniería. La Tabla 1 muestra la sección del marco que se ocupa del análisis de requisitos.

Desarrollo y trazabilidad de requisitos	
Porcentaje de requisitos que requieren reelaboración	15%
Número de requisitos con errores	750
Horas en reelaborar cada requisito	4,5
Horas ahorradas en corregir requisitos con errores después de la fase de requisitos	**3.375**

Tabla 1. Cálculo del marco de ROI del número de horas de ingeniería ahorradas cuando se corrigen requisitos erróneos con antelación.

Ahorro durante las pruebas

Las ineficiencias de la fase de pruebas también se capturan en este marco y suponen la mayor parte del ahorro. Con el diseño basado en modelos, se pueden generar pruebas automáticamente para verificar y validar modelos, y obtener una cobertura de pruebas completa. Este enfoque ahorra un tiempo valioso que de otro modo se dedicaría a solucionar el déficit de cobertura de pruebas en una etapa avanzada del proceso de desarrollo. Estos casos de prueba se pueden utilizar en varias etapas de pruebas, desde simulación en escritorio hasta pruebas de hardware-in-the-loop. También se pueden generar informes automáticamente que resuman los resultados de las pruebas y cumplan con los estándares de la industria. En un proyecto con 5.000 requisitos, por ejemplo, utilizar el diseño basado en modelos puede suponer un ahorro de hasta 12.000 horas, debido a la generación de casos de prueba exhaustivos durante la fase de modelado y la posibilidad de reutilizar casos de prueba en varios entornos de prueba. La Figura 5 muestra la sección del marco que se ocupa de evaluar las pruebas.

Pruebas y generación de informes	
Horas en solucionar el déficit de cobertura de pruebas por requisito	2
Factor de ahorro de generación de pruebas automática	70%
Horas ahorradas en solucionar el déficit de cobertura	**7.000**
Horas dedicadas a validar/depurar pruebas en laboratorio por requisito	2
Factor de reutilización de pruebas (desde pruebas en escritorio hasta laboratorio)	50%
Horas ahorradas en pruebas de laboratorio (reutilización de pruebas)	**5.000**

Tabla 2. Cálculo del marco de rentabilidad sobre la inversión correspondiente al número de horas de ingeniería ahorradas en el flujo de trabajo de pruebas e informes utilizando el diseño basado en modelos.

Ahorro total en desarrollo

El marco contiene varias secciones adicionales que se ocupan de diferentes ineficiencias de cada etapa del proceso de desarrollo. En resumen, en este ejemplo, casi la mitad del ahorro se produjo en la fase de pruebas (Figura 4). Esto se debe a un análisis de requisitos más exhaustivo en una etapa inicial del proceso de desarrollo, que genera menos errores en la fase de pruebas, cuando revisar y mejorar requisitos y pruebas es costoso. En pocas palabras, mejores requisitos generan mejores diseños. Las pruebas en etapas iniciales y continuas permiten identificar y solucionar más errores en la fase en la que se producen, lo que deja menos errores latentes en el software y reduce los costes totales de desarrollo.

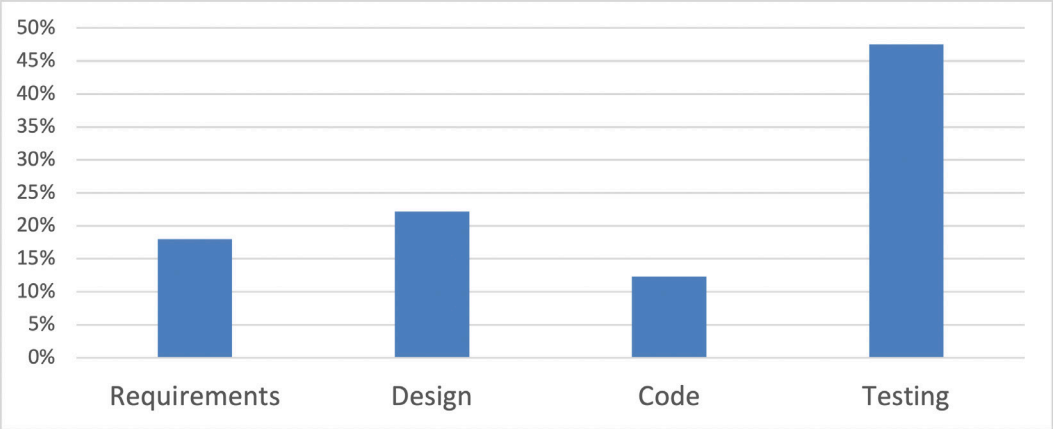


Figura 4. Porcentaje de ahorro total por fase de desarrollo.
Los ahorros en la fase de prueba representan la mayor parte del ahorro total.

Cuando las empresas colaboran con MathWorks en la adopción del diseño basado en modelos, el marco de rentabilidad sobre la inversión sirve como ayuda y orientación en el proceso de adopción, y permite identificar áreas que se beneficiarán de manera inmediata y significativa.

Ahorros más allá del ciclo de desarrollo

Si bien el marco de rentabilidad sobre la inversión cuantifica el ahorro de costes a lo largo del ciclo de desarrollo, este ahorro no es el único beneficio financiero de adoptar el diseño basado en modelos. El ritmo de desarrollo más rápido acelera el plazo de comercialización, lo que otorga una ventaja competitiva a los clientes que utilizan el diseño basado en modelos. Además, la reducción de gastos generales y errores de comunicación liberan recursos para las actividades comerciales principales y la innovación.

Resumen

Para la mayoría de las empresas, invertir en tecnologías y procesos conlleva cierto riesgo. El cálculo de la rentabilidad sobre la inversión que se describe en este white paper tiene como objetivo proporcionar pruebas analíticas que respalden la inversión en el diseño basado en modelos. Además de justificar la inversión, el marco de rentabilidad sobre la inversión permite identificar las áreas en las que el diseño basado en modelos proporciona el mayor ahorro, así como aquellas en las que una mayor investigación podría conducir a sustanciales reducciones adicionales de costes.

Para obtener un cálculo de la rentabilidad sobre la inversión personalizado para su equipo de trabajo u organización, ***comuníquese con ventas***.

Más información

- ***¿Por qué adoptar el diseño basado en modelos?***
- ***Cómo los equipos de ingeniería adoptan el diseño basado en modelos***
- ***Simulink para la simulación y el diseño basado en modelos***
- ***Evaluación de procesos y marco de madurez del diseño basado en modelos***

Referencias

1. Krasner, Jerry. How Product Development Organizations Can Achieve Long-Term Cost Savings Using Model-Based Systems Engineering (MBSE). (Cómo las organizaciones de desarrollo de productos logran ahorros en costes a largo plazo con MBSE). Octubre de 2015. ***https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:how_product_development_organizations_can_achieve_long-term_savings_1_.pdf***
2. MathWorks, OHB desarrolla software de orientación, navegación y control por satélite para vuelo en formación autónomo. ***https://www.mathworks.com/company/user_stories/ohb-develops-satellite-guidance-navigation-and-control-software-for-autonomous-formation-flying.html***
3. MathWorks, BAE Systems logra reducir en un 80% el plazo de desarrollo de radio definida por software. ***https://www.mathworks.com/company/user_stories/bae-systems-achieves-80-reduction-in-software-defined-radio-development-time.html***
4. Over, James W., *Managing Software Quality with the Team Software Process* (Gestión de la calidad del software con la metodología Team Software Process), Instituto de Ingeniería de Software. ***http://c-spin.net/2010/c-spin201004Managing%20Software%20Quality%20with%20the%20Team%20Software%20Process.pdf***
5. Vallespir, Diego, y William Nichols. *Analysis of Code (and Design) Defect Injection and Removal in PSP* (Análisis de inyección y eliminación de errores de código y diseño en PSP). UNIV. CARNEGIE MELLON, PITTSBURGH, PA, EE.UU., 2012. ***https://resources.sei.cmu.edu/asset_files/Presentation/2012_017_001_298088.pdf***
6. Tom King, Joe Marasco, *What Is the Cost of a Requirement Error?* (¿Cuál es el coste de un error de requisitos?) StickyMinds. ***http://www.stickyminds.com/sitewide.asp?ObjectId=12529&Function=edetail&ObjectType=ARTCOL***